# CipherShed - Bug #109

## Open Crypto Audit Project TrueCrypt CS-TC-3 - Keyfile mixing is not cryptographically sound

04/05/2015 02:48 AM - Jason Pyeron

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 04/05/2015 |
| **Priority:** | High | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | **Spent time:** | 0.00 hour |

**Description**

TARGETS: Use of Keyfiles in TrueCrypt volume passwords
DESCRIPTION: TrueCrypt allows the use of Keyfiles that are included with the user's passphrase in the derivation of the key used to unlock a volume. However, TrueCrypt does not mix the keyfile content into the passphrase in a cryptographically sound manner.
A 64-byte buffer is constructed, initially zero, called the keypool that is used to hold the entropy generated from the keyfiles. For each keyfile, a maximum of 1024 Kilobytes are read. A CRC (initially 0xFFFFFFFF and using the polynomial 0x04c11db7) is constructed, and for each byte in the file it is updated. Each time the CRC is updated, its four bytes are individually added into the keypool, modulo 256, and advancing (so the first time it updates bytes 0-3, the second time 3-7, and so on, wrapping around when it reaches 64.) The keypool output at the end of the first keyfile is used as the input keypool for the second keyfile.
After all of the keyfiles are processed, each keypool byte is added (modulo 256) into the user's password byte at that position. If the password is less than 64 bytes, the keypool byte in that position is used directly.
The use of CRC in this way is not cryptographically sound. When mixing entropy from multiple sources, an attacker who controls one source of entropy should not be able to fully negate or manipulate the other sources, even if the attacker is aware of what the other data is. [A previous example [https://defuse.ca/files2/poc/pocorgtfo03.pdf] demonstrating this flaw is a backdoor in the RDRAND instruction on older Linux kernels.] The use of a cryptographic
hash function is the correct way to mix entropy together – assuming the hash function is unbroken, the best attack able to be mounted is a brute-force search for an input that, when combined with the uncontrolled input, yields a desirable output.
In the current implementation an attacker is able to calculate the resulting keypool following the uncontrolled keyfiles, and then (because of the use of CRC) calculate a keyfile that will entirely negate the established pool. If an attacker manipulates the keypool to be all 0x00, it will be as if no keyfiles were used at all.
Recommendation: Use a cryptographic hash function (possibly in an HMAC construction) to prevent an attacker from manipulating a keyfile that could be used to negate the use of other keyfiles. When using novel cryptographic techniques, clearly document the design of the approach in a separate document and encourage review by the professional and academic community.
Note: After completing the review and documenting this bug, CS was alerted to its previous discovery by the Ubuntu Privacy Remix Team in 2011 [https://www.privacy-cd.org/downloads/truecrypt_7.0a-analysis-en.pdf].
12A